VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

2012/2013 **Tomáš Hric**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Tomáš Hric**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: netdevelo s.r.o.
2. Struktura závěrečné zprávy:
a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta
b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti
c) Zvolený postup řešení zadaných úkolů
d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe
e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe
f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vedl odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Mgr. Marek Menšík, Ph.D.**

Konzultant bakalářské práce: Ing. Lukáš Heinz

Datum zadání: 16.11.2012
Datum odevzdání: 07.05.2013

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 13. 6. 2013

Tomáš Hric

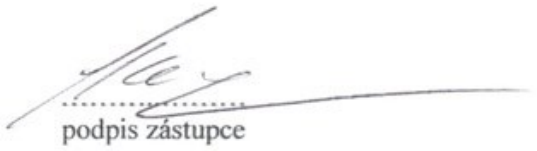
.....

podpis studenta

Prohlášení zástupce spolupracující právnické nebo fyzické osoby

„Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských/magisterských programech VŠB-TU Ostrava.“

Dne: 29.5.2013



podpis zástupce

Abstrakt

Tato práce popisuje mé působení ve společnosti netdevelo s.r.o., která se primárně věnuje vývoji internetových obchodů přizpůsobených na míru. Praxi jsem vykonával na pozici programátora na oddělení realizace projektů. V rámci odborné praxe jsem pracoval na interních aplikacích firmy. U dvou aplikací, kterým jsem se věnoval téměř celé období praxe, jsem měl provést analýzu požadavků, návrh a následnou implementaci. První aplikace „Pošťák“ je určena pro evidenci příchozí a odchozí papírové pošty. Druhá aplikace „Pohledávky“ slouží k snadnému vymáhání pohledávek od klientů.

Klíčová slova

webová aplikace, PHP, Nette Framework, databáze, aplikace Pošťák a Pohledávky, MVP/MVC architektura

Abstract

This bachelor thesis describes my career in the company netdevelo Ltd. which is primarily specialized in development of customer-tailored e-shops. I worked as programmer in the department Implementation of projects. Within this professional practice I worked on the internal applications of the company. I spent almost all the time of my practice to analyze the requirements, to make a design and subsequent implementation of two applications. The first application named “Pošťák” is designed for recording incoming and outgoing paper mail. The second application named “Pohledávky” is made for easy debt recoveries from clients.

Key words

web application, PHP, Nette Framework, database, applications Pošťák and Pohledávky, MVP/MVC architecture

Seznam použitých zkratek

Zkratka	Anglický význam	Význam
MVP	Model-View-Presenter	Architektura aplikace
MVC	Model-View-Controller	Architektura aplikace
PHP	Hypertext preprocessor	Scriptovací programovací jazyk
DB	Database	Databáze
CMS	Content Manager System	Systém pro správu obsahu
SEO	Search Engine Optimization	Optimalizace pro vyhledávače
Ajax	Asynchronous JavaScript and XML	Skupina technologií umožňující měnění obsahu stránky bez nutnosti znovunačtení
CSV	Comma-separated values	Jednoduchý souborový formát tabulkových dat
SQL	Structured Query Language	Strukturovaný dotazovací jazyk
SVN	Subversion	Systém pro správu a verzování zdrojových kódů

Seznam použitých termínů

Termín	Význam termínu
Copywriting	Tvůrčí činnost pro psaní poutavých textů, které mají za účel zaujmout čtenáře a prodat produkt nebo službu zákazníkovi.
Cron	Je Linux/Unix nástroj, který spouští periodicky v určitý čas nějaký proces (script).
Wireframe	Drátěný model definující funkci a obsah aplikace.

Obsah

1 Popis odborného zaměření firmy

1.1 O společnosti

Firma netdevelo s.r.o. se zabývá vývojem profesionálních internetových aplikací již od roku 2003. Její vlajkovou lodí jsou internetové obchody na platformě ShopSys. V roce 2007 byla firma zapsána do obchodního rejstříku. Nabízí konzultace v oblasti elektronické komerce. Hlavní sídlo firmy je v Ostravě, pobočka na Slovensku a jedno kontaktní místo v Praze. Zaměstnává více než 30 zaměstnanců. Dále nabízí následující služby:

- Internetové aplikace a objednávkové portály
- CMS
- Copywriting, SEO optimalizace a analýza
- Tvorba grafiky, reklamy a bannerů
- Hosting a provoz serverů

Hlavním produktem je internetový obchod ShopSys, kterých se implementovalo kolem 500 kusů. ShopSys není úplně krabicovým produktem, jak by se mohlo zdát, ale je upravován na míru podle požadavků klienta. Při takovém počtu projektů a zaměstnanců je nutné firmu spravovat přes interní aplikace. Některé aplikace jsem v průběhu praxe upravoval a jiné vytvořil.

1.2 Popis pracovního zařazení

Do společnosti netdevelo s.r.o. jsem byl přijat na oddělení realizace projektů. Podílel jsem se na úpravách a vytváření interních aplikací. Při vytváření aplikací jsem si prošel několika fázemi softwarového procesu – analýza požadavků, návrh a především implementace. Nejvíce času jsem však strávil na pozici PHP programátora při implementaci a úpravách kódů.

1.3 Firemní interní aplikace

Všechny aplikace používané ve firmě jsou postaveny na oblíbené kombinaci technologií PHP a MySQL. Jsou hostované na firemních serverech. Při tvorbě nových aplikací Pošťák a Pohledávky musela volba padnout právě na tyto technologie. Znalost HTML, CSS a JavaScriptu byla zde samozřejmostí.

2 Úkoly zadané v průběhu odborné praxe

2.1 Úpravy aplikace Develtesty

Prvním zadaným úkolem po příchodu do firmy bylo provést několik potřebných úprav na aplikaci Develtesty. Ta složí k zaznamenávání chyb nalezenými testery.

Aplikace byla napsána v PHP framerworku Yii, který má MVC architekturu. Neznalost tohoto frameworku značně komplikovalo úpravy a rozšíření aplikace podle pokynů. Takže menší část času zabralo prostudování oficiální dokumentace v angličtině.

Jednalo se o opravy špatně ukládaných záznamů, odesílání prázdných emailů a jiné logické a běhové chyby aplikace. Další výraznou úpravou byla optimalizace rychlosti načítání. Prvně jsme se zaměřili na optimalizaci SQL dotazů a následně optimalizaci PHP kódu. Program také generoval velký výstup s tisíci řádky javascriptového kódu, který se podařilo obecně zapsat na pár řádků. Abychom nemuseli přepisovat celou aplikaci, použili jsme v některých případech ajaxové načítání obsahu. Jiné změny se týkaly rozšíření – např. přidání boxu pro zobrazování náhledu chyby po najetí myší, aby se nemusel detail otevírat v novém okně apod.

2.2 Aplikace Pošťák

Dalším úkolem bylo vytvořit novou aplikaci, se kterou mají pracovat asistentky firmy. Při realizaci jsme postupovali přibližně podle vodopádového modelu. Prošel jsem si tedy několika fázemi softwarového procesu. V první fázi jsem provedl sběr a analýzu požadavků. Při sběru požadavků jsem se od asistentek dozvěděl, že se má jednat o software pro evidenci příchozí a odchozí klasické pošty, která se původně musela ručně zapisovat do knihy. Nová aplikace navíc měla být propojena s firemní databází klientů – jelikož většina poštovní komunikace probíhá mezi firmou a klientem (ale nejen s klientem). Dále podle specifikace bylo nutné ukládat o poště informace jako je způsob doručení, směr, datum, cenu zásilky, jméno vyřizujícího konzultanta či jiné osoby, text a případně projekt, na který se pošta vztahuje. Každá pošta musela jít zařadit do kategorie (např. dodatky, dohody, smlouvy, protokoly apod.) a navíc každá z těchto kategorií má několik dalších podkategorií (nebudu uvádět), u které je nutné opět evidovat údaje určené pro danou podkategorii. Každá kategorie a podkategorie má další vstupní data. Dalším požadavkem bylo upomínat označenou poštu emailem, jenž se má zasílat v určitých časových intervalech na email asistentky. Samotný výpis seznamu pošty vyžadoval filtrování podle uvedených sloupců a kategorií. Aplikace dále měla umožnovat tisk pošty za určité časové období. Nejdůležitější funkce aplikace jsou:

- Vkládat, upravovat a mazat poštu
- Rozdělovat poštu do speciálních kategorií

- Napojit na aplikaci Klient, kde je databáze všech klientů
- Výpis pošty s filtrováním
- Upomínat poštu emailem
- Tisk pošty

Po dokončení analýzy jsem vytvořil wireframe aplikace a odprezentoval asistentkám k odsouhlasení. Po úspěšném schválení jsem započal třetí fázi procesu – implementace. Mezi nefunkční požadavky patřilo, že bude informační systém napsán jako webová aplikace postavena na jazyku PHP a databázi MySQL.

2.3 Aplikace Pohledávky

Aplikace Pohledávky byl název nové aplikace, kterou jsem měl vytvořit. Při vývoji se postupovalo zcela stejně jako u předchozí aplikace – analýza, návrh, schválení návrhu, implementace a nasazení. Během analýzy nebylo nikomu zcela jasné, jak bude program fungovat, ale jen to, co by měl umět. Zjistil jsem, že aplikace má usnadňovat práci projektovým konzultantům a asistentce při vymáhání pohledávek od klientů. Má se provádět export faktur po splatnosti z fakturačního systému. K faktuře chtěli zobrazovat informace jako je číslo faktury, datum vystavení, datum splatnosti, název firmy, projekt, projektový konzultant (PK), prodlení dnů, cena bez/s DPH a položky na faktuře. Při vyřizování pohledávek by měla být možnost napsat poznámku PK, poznámku asistentky, uvést smluvní pokutu, stav pohledávky nebo možnost předat pohledávku vymáhací firmě. Seznamy musely mít filtry, aby se v nich snadněji hledalo. Dále bylo třeba rozeznat, které pohledávky vymohla asistentka, neboť se z nich vypočítávají provize. Další funkcí aplikace Pohledávky byl výpis seznamu všech klientů, kteří jsou aktuálně vymáháni vymáhací agenturou třetí strany. U každého klienta se má zobrazit seznam jeho pohledávek, k nimž měl systém umožnit vyplnit datum předání, popis ze strany vymáhací firmy a umožnit vložení přílohy či zrušit vymáhání. Posledním požadavkem byl script, který by rozesílal všem konzultantům emaily s týdenním přehledem obsahující výpis nových pohledávek a pohledávek, které musí konzultant ještě vyřídit. Nejdůležitější funkce jsou:

- Stahování faktur z fakturačního systému
- Výpis aktuálních pohledávek s možností filtrování
- Editace pohledávek, psaní poznámek při vymáhání
- Výpis pohledávek vymožených asistentkou (filtr od – do) + výpočet provizí
- Výpis klientů předaných vymáhací firmě a seznam jejich pohledávek + nahrávání příloh s dokumenty
- Rozesílání emailů s týdenním přehledem

Podstatným úkolem bylo nalézt způsob, jak s danou aplikaci pracovat a tato úloha spadala na mne. Měl jsem velice volnou ruku při tvorbě návrhu aplikace. Pro vytváření mock-ups jsem použil software k tomu určený a opět ho odprezentoval ke schválení.

2.4 Ostatní úkoly

Mezi vývojem nebo po dokončení těchto větších aplikací jsem měl pracovat na menších úkolech, které se týkaly především programování. Jednalo se například o opravy na internetových obchodech ShopSys. Změny jsem ukládal do verzovacího systému SVN. Dále jsem upravoval interní aplikaci pro generování služebních cest. Vytvořil jsem také plugin do open-source softwaru MediaWiki, který firma používá pro informování zaměstnanců. Plugin měl na každé stránce zobrazovat všechny odkazované stránky na danou stránku. Ještě jedním požadavkem bylo upravit standartní vyhledávání MediaWiki a přidat našeptávání.

Mým posledním úkolem ve společnosti netdevelo s. r. o. bylo vytvořit miniaplikaci pro práci se statistikami. Statistiky v CSV souboru se měly importovat do aplikace, rozparsovat a uložit do databáze. Nad databází se mají provádět SQL dotazy, které aplikace měla spravovat. Výsledky dotazů se měly zobrazovat v tabulce s možností exportu do CSV souboru. Situaci značně komplikoval fakt, že importované soubory nemusely mít stejnou strukturu. Aplikace měla taky umožňovat smazat všechny záznamy z kteréhokoliv importu.

3 Postup řešení zadaných úkolů

Tato kapitola pojednává o mém přístupu k řešení problémů při vývoji aplikací Pošťák i Pohledávky, které jsem vyvíjel většinu času během praxe.

3.1 Zvolený framework

Po dokončení analýzy Pošťáka i Pohledávek jsem měl možnost si sám vybrat, zdali chci aplikaci naprogramovat v čistém PHP či zvolit nějaký framework. Volba padla na framework ze dvou důvodů. Prvním důvodem bylo, že může odstranit některé nedostatky, které PHP vlastní. Ku příkladu může omezit zbytečně velkou benevolentnost jazyka, a tak předejít zbytečnému hledání chyb při programování. Další výhodou je, že správný výběr frameworku zajistí, že aplikace bude naprogramována kompletně objektově. Díky objektově orientovanému přístupu jsem mohl uplatnit znalosti získané během studia. Druhý důvod byl ten, že u komplexnějších aplikací mohu využít rozšíření (práce se soubory, formuláři, cache, řetězci apod.), které framework nabízí.

3.1.1 Nette Framework

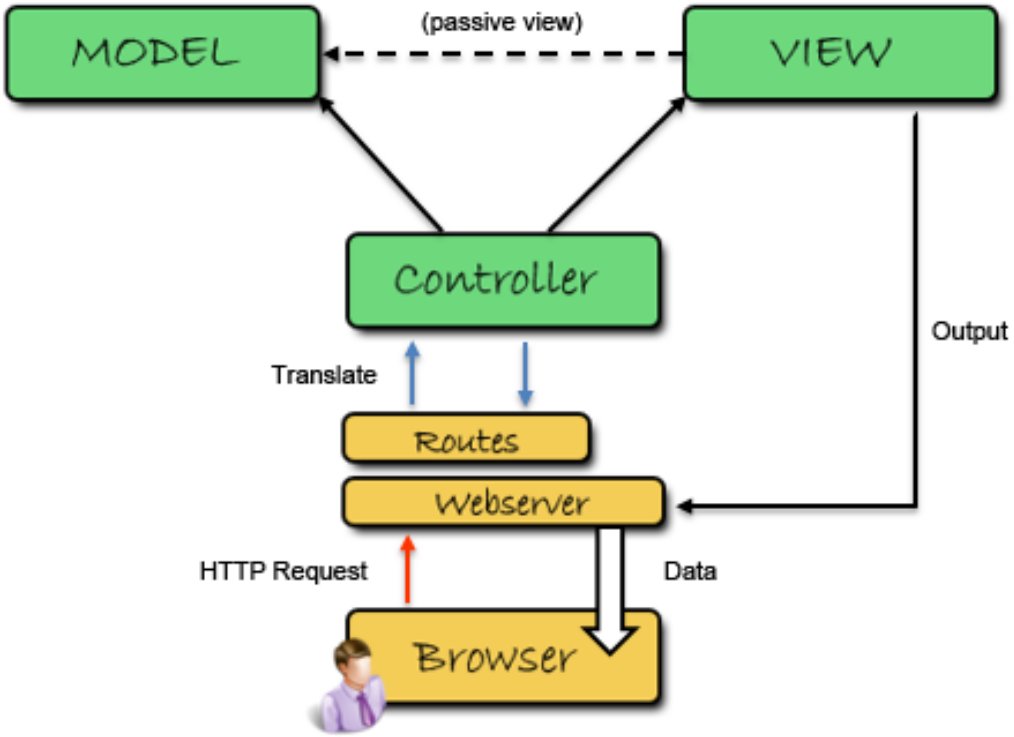
Pro programování aplikací jsem si zvolil Nette Framework 2.0 pro PHP 5.3+. Tato verze PHP už dovoluje používat jmenné prostory. V této bakalářské práci bych chtěl Nette věnovat jednu podkapitolu, jelikož jsem s tímto frameworkem během praxe pracoval téměř každý den a napsal v něm celkem 3 aplikace pro firmu netdevelo.

Nette Framework znatelně ulehčí práci při psaní programu. Disponuje čistým objektovým návrhem, takže aplikace bude mít přehlednější kód. Oproti čistému PHP vede k dobrým zvyklostem návrhu programu a navíc patří k nejrychlejším frameworkům vůbec. Architektura je rozdělena podle MVP vzoru (podobnost s MVC, viz kapitola 3.1.2). Důvody proč jsem si vybral Nette framework:

- **Šablonovací systém** – odděluje aplikační vrstvu od prezenční vrstvy. Podporuje mnoho maker a helperů. Rychlost je dobrá, jelikož se šablony překládají do nativního PHP kódu a ukládají se do cache.
- **Ladicí nástroj** – umožní rychle odhalit a opravit chyby, provádí logování, měření čas, výpis provedených SQL dotazů apod.
- **Práce s databází**
- **Bezpečnost** - obrana proti útokům XSS, CSRF, podstrčení session, aj.
- **Tvorba komponent** – komponenta představuje vykreslitelný objekt, umožní znovupoužitelnost kódu
- **Formuláře** – knihovna pro rychlé vytváření formulářů a jejich validaci
- práce s adresáři a soubory, routování adres, práce s emaily, a další knihovny

3.1.2 MVC a MVP architektonické vzory

Model-View-Controller vznikl z potřeby oddělit aplikaci na řídící logiku (controller), aplikační logiku (model) a zobrazující rozhraní (view). Tato separace vede k přehlednějšímu kódu a možnosti práce na jednotlivých částech odděleně (např. při práci v týmu). Obrázek znázorňuje architekturu běžných webových MVC aplikací:



Obrázek 3.1: Architektura webové aplikace

- **Router** – zpracovává http požadavek a vytváří překlad mezi URL a akcí controlleru/presenteru. U mnohých frameworků funguje obousměrně.
- **Controller/Presenter** – zpracovává uživatelské požadavky, na jejichž základě volá aplikační logiku (tj. model). Poté požádá view o vykreslení dat.
- **Model** – datový a funkční základ aplikace, který se stará o vytažení nebo změnu dat. Má dané rozhraní, prostřednictvím něhož ostatní části aplikace k modelu přistupují. O svém okolí neví nic.
- **View** – pohled, jenž obstarává vykreslení dat. U webových aplikací se většinou jedná o nějaký šablonovací systém, který zpracovává šablonu s html kódem.

Nette Framework se podle dokumentace přibližuje spíš k MVP vzoru, kde presenter pracuje přímo s view, takže je tato vazba mnohem silnější. Stejně jako u MVC, model se nezajímá o ostatní. V Nette dokonce view nemusí vědět o modelu, čili mezi nimi zmizí vazba. Jedná se o tzv. Passive view, které popsal Martin Fowler na svém webu.

Rozdíly mezi MVC a MVP zmiňuji z toho důvodu, že jsem během praxe pracoval s oběma vzory ve frameworku Yii a Nette.

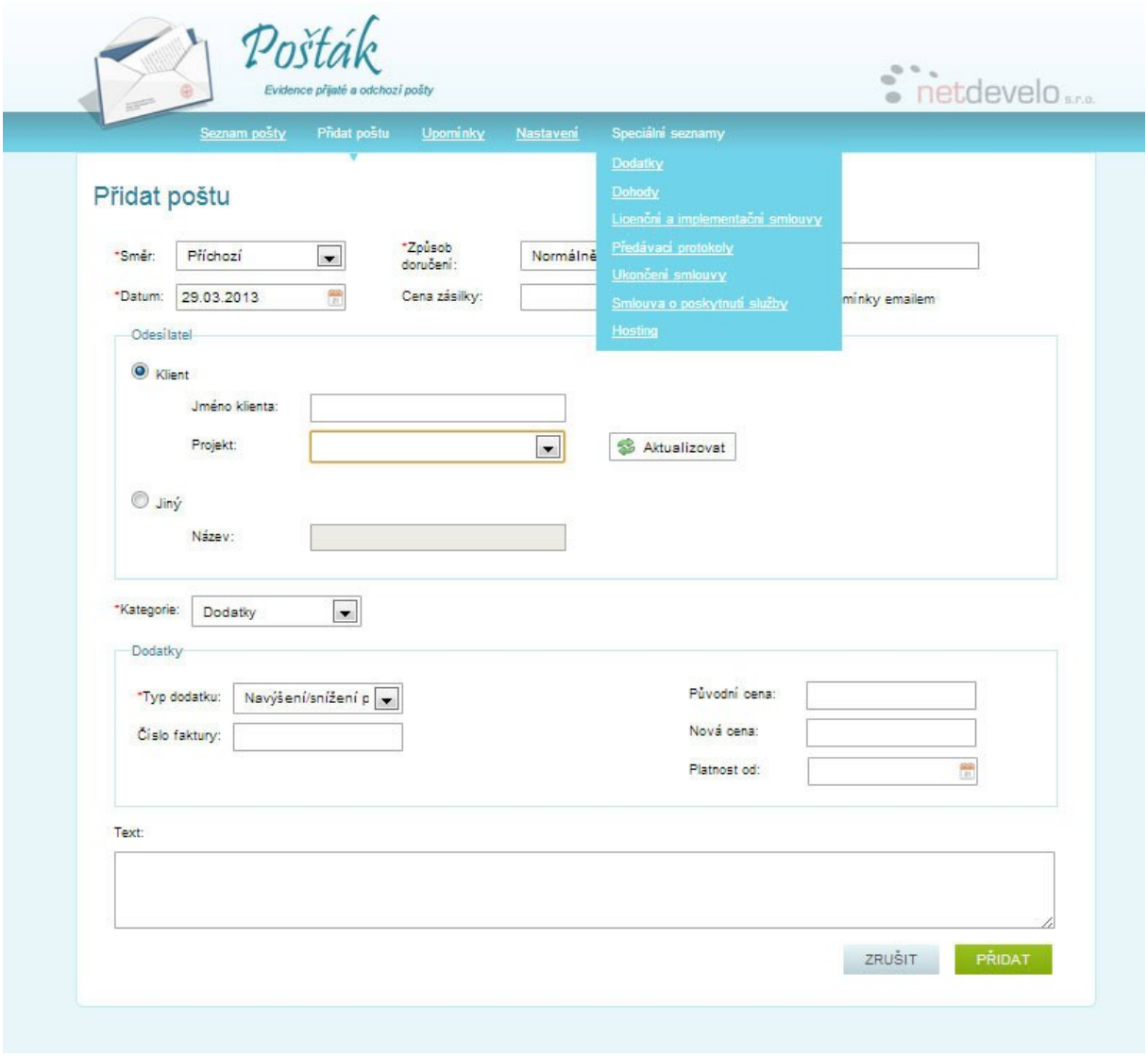
3.2 Postup při vývoji informačního systému Pošťák

Jak jsem zmínil v druhé kapitole, při vývoji softwaru se postupovalo přibližně podle vodopádového modelu, který v roce 1970 poprvé popsal Winston W. Royce ve svém článku. S kolegou z VŠB, který také absolvoval odbornou praxi, jsme provedli analýzu a každý zvlášť jsme vytvořili návrh aplikace (wireframes). Nakonec jsme spojili to nejlepší z obou návrhů do finálního, jenž jsme následně odprezentovali ke schválení.

Programování aplikace jsem prováděl již samostatně. Prvním krokem byl návrh datového relačního modelu, jelikož bylo zapotřebí si ujasnit, která data se budou stahovat z aplikace Klient. V aplikaci Klient jsou totiž informace o všech klientech společnosti netdevelo. Zde nastával problém, jak pracovat s daty, která nejsou ani fyzicky na stejném serveru. Nejjednodušším řešením bylo vytvořit na databázovém serveru (s Klientem) view, prostřednictvím něhož se budou pravidelně stahovat data do druhé databáze. Stahování dat zařizuje Cron script spouštěný automaticky každý den.

Dalším Cronem byl script na zasílání emailových upomínek asistentkám. Email se má odesílat po určitém počtu dnů uvedeném v nastavení. Přesto se musí script spouštět každý den, jelikož neví, kdy byl email naposledy odeslán.

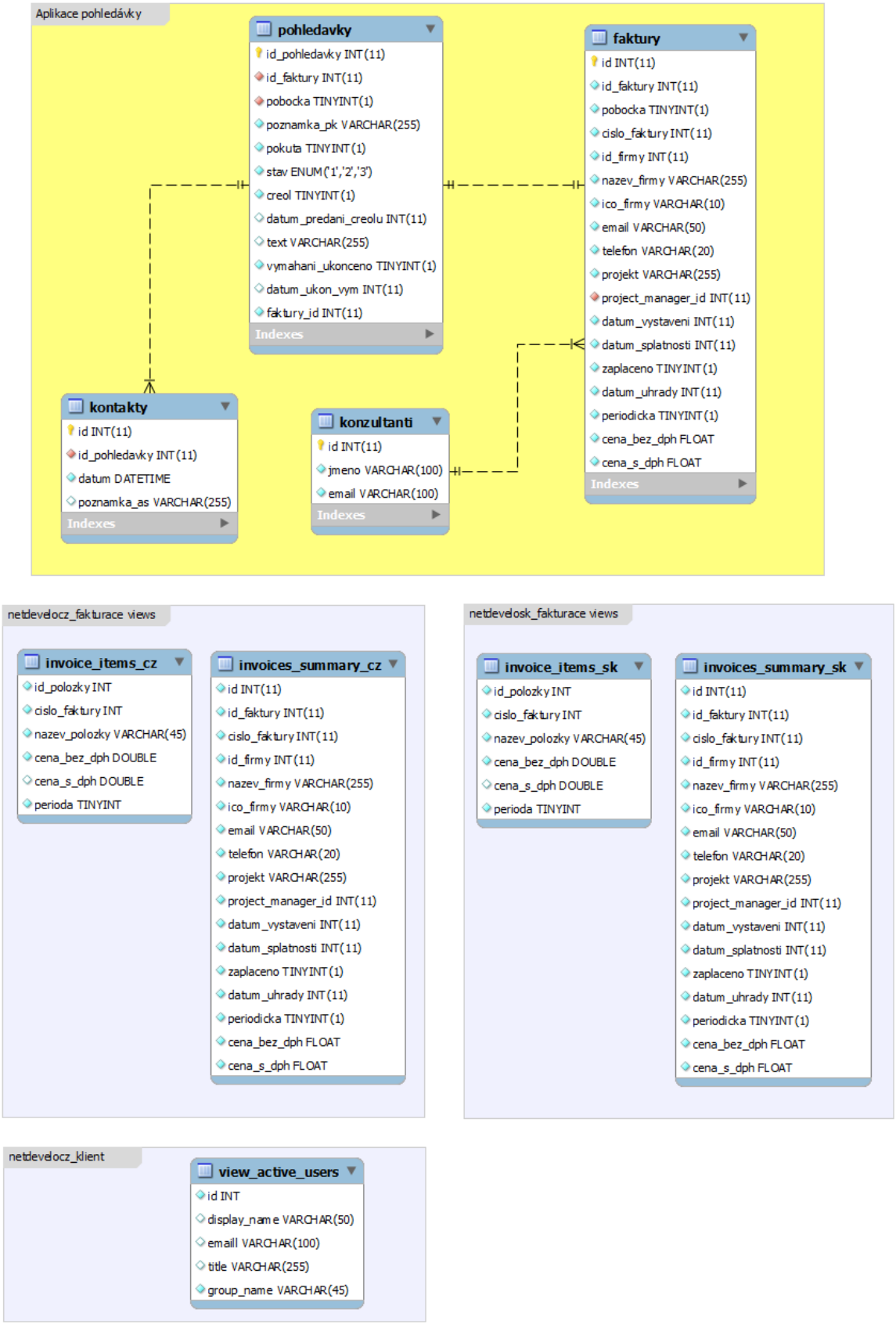
Informační systém jsme se snažili navrhnout tak, aby byl uživatelský přívětiví, snadno ovladatelný a připraven na každodenní rutinní práci při zadávání dat. Konkrétním příkladem může být dynamicky formulář s postupným sestavováním. Dalším příkladem je výběr klienta (odesílatele/příjemce), kde bylo nemyslitelné vložit combobox, protože při tak velkém počtu klientů, by bylo pro uživatele nepohledné hledat klienta v dlouhém seznamu. Toto jsem vyřešil obyčejným textboxem s funkcí autocomplete a pomocí hidden inputu. Autocomplete (česky našeptávání) jsem napsal v javascriptu, který při psaní do textboxu dělá ajaxové požadavky na server a ten vrací výsledky v JSON formátu. Výsledky jsou uživatelsky zobrazeny a po vybrání klienta uživatelem se vloží ID klienta do hidden inputu. Do databáze se totiž ukládá Id, nikoli jméno klienta.



Obrázek 3.2: Ukázka formuláře pro přidání pošty

3.3 Postup při vývoji informačního systému Pohledávky

Prvním krokem byl opět návrh ERD modelu. Abych mohl vypsat aktuální pohledávky, potřeboval jsem přístup do databáze fakturačního systému, která je bohužel na jiném serveru. Fakturační systémy jsou dokonce dva – český a slovenský. Abych mohl provádět dotazy nad všemi tabulkami zároveň, potřeboval jsem je dostat na jeden server. Pro každou DB fakturačních systémů jsem vytvořil jedno view, ze kterého stahuje Cron do lokální databáze aplikace každý den všechny faktury. Díky view můžu jednoduše přenášet jen sloupečky (informace o faktuře), které potřebuji. Obě view z fakturačních systémů ve výsledku spojuji do jedné tabulky. Situaci demonstruje obrázek 3.2. Původně jsem si myslel, že budu přenášet pouze nezaplacené faktury po vypršení data splatnosti, jenomže aplikace měla navíc umět zobrazit faktury vymožené asistentkou. Takže by faktury po zaplacení zmizly. Položky faktury není třeba kopírovat, ale stačí vytvořit druhé připojení na server, jelikož není potřeba provádět spojování tabulek nebo vnořené dotazy s druhou databází.

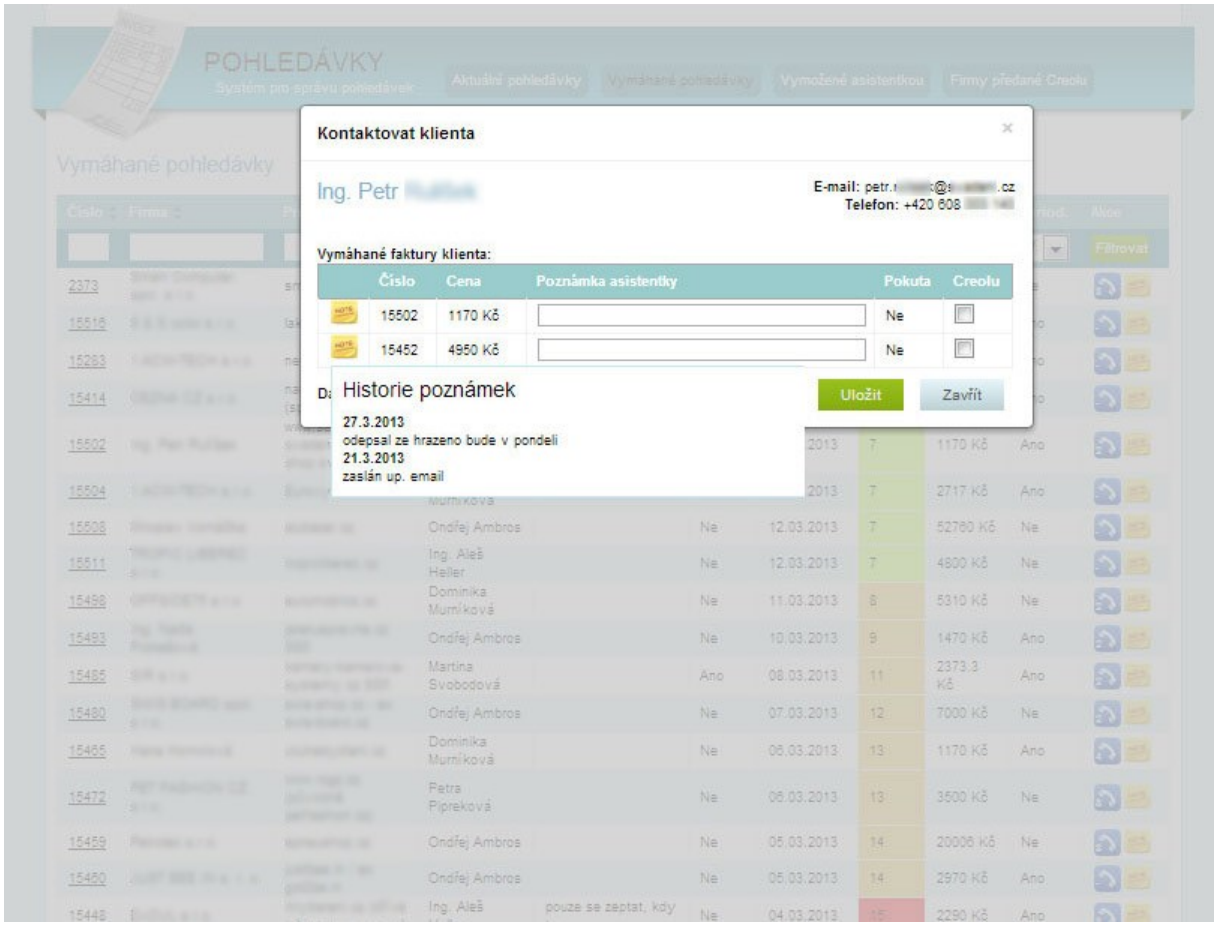


Obrázek 3.3: ER Diagram – aplikace Pohledávky

Měl jsem velkou volnost při návrhu aplikace, takže jsme vymysleli pár zlepšení. Oddělil jsem práci konzultantů a asistentky pomocí záložek (nikoliv na uživatelské účty). Pro kontaktování klienta jsem vytvořil speciální okno s kontaktními údaji a možnosti zapisování poznámek asistentky do historie. Díky tomu může asistentka vidět, o čem se s klientem bavila při minulé konverzaci.

Moderní webové aplikace dnes vyžadují skupinu technologií ajax. Ajax značně dokáže usnadnit práci s aplikací. Příkladem může být jeden z požadavků konzultanta, který chtěl implementovat tzv. řádkovou editaci. Jedná se o úpravu záznamu přímo v tabulce se záznamy. Po dvojkliku na záznam se z řádku stane formulář a kliknutím mimo něj se formulář uloží. Práci také zpohodlní modální vyskakovací okna, díky nimž nemusí uživatel znovu načítat stránku a vracet se zpět. Tyto okna jsem použil například pro zobrazení položek faktury, zobrazení historie poznámek, seznamu příloh apod.

Další poměrně zajímavou částí aplikace byl výpočet provizí z pohledávek, které vymohla asistentka. Provize se spočítá z celkové ceny položek všech faktur. Cena však nejde jednoduše sečíst, protože některé položky jsou v Eurech a je nutné je převést podle kurzovního lístku. Kurz měny je scriptem zjišťován z API webu České národní banky, rozparsován a ukládán do cache paměti s expirací 24 hodin.



Obrázek 3.4: Ukázka seznamu pohledávek a kontaktního formuláře

4 Teoretické znalosti a dovednosti uplatněné v průběhu odborné praxe

4.1 Znalosti získané v průběhu studia

Pro práci ve společnosti netdevelo s.r.o. jsem uplatnil především předměty zaměřené na programování. Přestože se jazyk PHP na škole v době mého studia nevyučoval, nebyl problém se jej doučit samostatně. Dobré návyky a zvyklosti objektově orientovaného programování jsem si převzal z jazyka C# z předmětu Programovací jazyky II.

Značným přínosem mi byly předměty Úvod do databázových systémů a navazující předmět DAIS, které mne naučily dobře navrhovat databáze a informační systémy. Během praxe jsem musel konstruovat i složitější SQL dotazy s ohledem na rychlost vykonávání.

Při analýze požadavků jsem uplatnil znalosti získané při absolvování předmětu Vývoj informačních systémů pro sběr požadavků, abych navrhl systém, který bude co nejlépe splňovat představy zadavatele. Dále mě předmět seznámil s návrhovými vzory, se kterými jsem se během implementace setkal. Především vzory MVC a Active Record.

4.2 Scházející znalosti

Aplikace Develtesty byla napsaná ve frameworku Yii, se kterým jsem se do té doby ještě nesetkal. Proto provádět optimalizační úpravy nebylo zrovna jednoduché. Dále jsem poprvé začal používat verzovací systém SVN, které pro mě předtím pozbýval významu. Znatelně šly poznat chybějící praktické zkušenosti při analýze požadavků.

5 Používané technologie a nástroje

Základním programovacím jazykem byl scriptovací jazyk PHP 5.3. V některých případech starší verze PHP 5.2. Jako SŘBD jsem používal MySQL (systém řízení báze dat). Mnohokrát jsem využíval javascriptovou knihovnu jQuery i v souvislosti s ajaxem. V PHP jsem použil například knihovnu Dibi pro práci s databázovou vrstvou nebo šablonovací systém Latte, který je součástí Nette Frameworku. Pro psaní šablon jsem potřeboval znát značkovací jazyk HTML a kaskádové styly.

U některých aplikací bylo vyžadováno veškeré změny ukládat to repozitáře SVN. Toto umožní dohledání autorů kódů a popis provedených změn.

Dále jsem používal kromě programátorského IDE také nástroje pro tvorbu wireframes nebo software pro tvorbu ER diagramů, abych mohl řešení předvést ke schválení.

6 Dosažené výsledky v průběhu praxe

Časově nejnáročnější bylo vytvořit aplikace Pošťák a Pohledávky, na kterých jsem pracoval téměř celou praxi. Obě aplikace jsou dnes používané každý den zaměstnanci firmy. Svůj účel splňují do posledního požadavku a doufám, že ve velké míře ulehčují práci zaměstnancům. Všechny ostatní zadané úkoly se mi také podařilo splnit.

Celá praxe pro mě byla přínosná po odborné stránce a to i díky některým zaměstnancům, kteří mi byli nápomocni při řešení vzniklých problémů. Měl jsem také možnost se účastnit zajímavých přednášek z oboru. Jsem rád, že jsem získal další odborné kompetence a že jsem měl možnost sledovat, jak probíhá vývoj větších projektů a jak se spolupracuje v týmu.

Použitá literatura

[1] NETTE FRAMEWORK. *Dokumentace.* [online]. [cit. 2013-03-24]. Dostupné z: http://doc.nette.org/cs/

[2] VONDRÁK, Ivo. *Úvod do softwarového inženýrství*. Ostrava, 2002 [cit. 2013-04-14]

[3] MARTIN FOWLER. *Passive View.* [online]. [cit. 2013-04-21]. Dostupné z: http://martinfowler.com/

[4] GUTMANS, Andi. *Mistrovství v PHP 5*. Vyd. 2. Brno: Computer Press, 2007, 655 s. ISBN 978-80-251-1519-0.

[5] NETDEVELO S.R.O. *Netdevelo.* [online]. [cit. 2013-03-24]. Dostupné z: http://www.netdevelo.cz/